RRT-Based Nonholonomic Motion Planning Using Any-Angle Path Biasing

Luigi Palmieri Sven Koenig Kai O. Arras

***Abstract*— RRT and RRT* have become popular planning techniques, in particular for high-dimensional systems such as wheeled robots with complex nonholonomic constraints. Their planning times, however, can scale poorly for such robots, which has motivated researchers to study hierarchical techniques that grow the RRT trees in more focused ways. Along this line, we introduce Theta*-RRT that hierarchically combines (discrete) any-angle search with (continuous) RRT motion planning for nonholonomic wheeled robots. Theta*-RRT is a variant of RRT that generates a trajectory by expanding a tree of geodesics toward sampled states whose distribution summarizes geometric information of the any-angle path. We show experimentally, for both a differential drive system and a high-dimensional truck-and-trailer system, that Theta*-RRT finds shorter trajectories significantly faster than four baseline planners (RRT, A*-RRT, RRT*, A*-RRT*) without loss of smoothness, while A*-RRT* and RRT* (and thus also Informed RRT*) fail to generate a first trajectory sufficiently fast in environments with complex nonholonomic constraints. We also prove that Theta*-RRT retains the probabilistic completeness of RRT for all small-time controllable systems that use an analytical steer function.**

I. Introduction

Any-angle search is a family of discrete search techniques which, unlike A* or Dijkstra's algorithm, find paths that are not constrained to grid edges. Daniel *et al.* [1] introduce Theta*, an any-angle search technique whose paths are only slightly longer than true shortest paths. The authors show that the basic variant of Theta* finds shorter paths than Field D*, A* with post smoothing and A* on grids, see Fig. 2. Rapidly exploring Random Trees (RRT) [2] is a sampling-based motion planner that expands trees in the state space toward newly sampled states. An optimal variant, RRT* [3], rewires the trees based on the notion of cost. To improve the performance of sampling-based motion planners, recent research has combined them with discrete search techniques [4, 5, 6, 7]. None of these studies, however, combine any-angle search with RRT variants although its properties (such as finding shorter paths than A* with fewer heading changes) are likely beneficial for the performance of the combination.

L. Palmieri and K.O. Arras are with the Social Robotics Lab, Dept. of Computer Science, University of Freiburg, Germany, {palmieri,arras}@cs.uni-freiburg.de. Kai Arras is also with Bosch Corporate Research.

S. Koenig is with the IDM Lab, Dept. of Computer Science, University of Southern California, USA, skoenig@usc.edu.

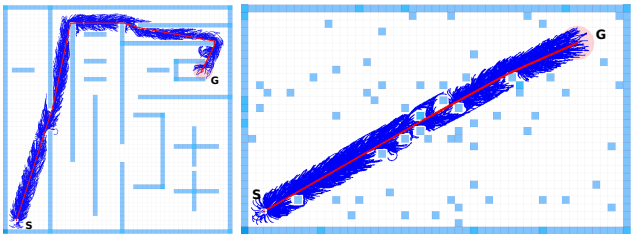

Fig. 1. Theta*-RRT trees in two example environments used in the experiments. **Left:** *Maze* environment. **Right**: *Random map* environment. The trees (in **blue**) grow smoothly towards the goal in a subspace centered around the any-angle path (in **red**).

In this paper, we present Theta*-RRT with the objective of rapidly generating smooth and short trajectories for high-dimensional nonholonomic wheeled robots. Theta*-RRT is a hierarchical technique that combines (discrete) any-angle search with (continuous) RRT motion planning. It improves the efficiency of RRT in high-dimensional spaces substantially by transferring the properties of the any-angle path to the final trajectory. Theta*-RRT considers a continuous control space during planning: It uses steer functions instead of random control propagations to exploit as much knowledge of the nonholonomic constraints of the system as possible and to ensure both high planning efficiency and high trajectory quality. Since heuristics can also be misleading and degrade planning performance, we prove that Theta*-RRT retains the probabilistic completeness of RRT for all small-time controllable systems that use an analytical steer function. We evaluate the approach using a 3D differential drive robot and a 8D truck-and-trailer system. We compare it to four baseline planners: RRT, RRT* (and thus also Informed RRT* [8] which behaves like RRT* until a first trajectory is found), A*-RRT, and A*-RRT* [7]. The evaluation shows that Theta*-RRT is significantly faster and produces shorter high-quality trajectories than those of the baselines.

The paper is structured as follows: We describe related work in Sec. II and Theta*-RRT in Sec. III. We present experiments in Sec. IV and discuss their results in Sec. V. Probabilistic completeness of Theta*-RRT is proven in Sec. VI.

II. Related Work

Prior research has combined discrete search with continuous sampling-based motion planning. For example, Plaku *et al.* [4, 5] propose a planner where a search-based planner finds a sequence of decomposition regions

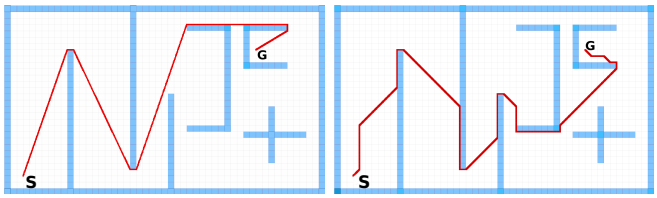

Fig. 2. Comparison of Theta* and A*. **Left:** The any-angle path of Theta* (in **red**) is not constrained to grid edges. **Right**: The grid path of A* (in **red**) is constrained to grid edges and part of a different homotopy class. It is longer and has more heading changes.

that are then used to guide how RRT grows the tree. Bekris and Kavraki propose the Informed Subdivision Tree technique [6] that uses a heuristic to direct the tree growth and improve the coverage of the state space. In contrast to these two planners, Theta*-RRT biases the tree growth mainly in the homotopy class found by Theta* and considers a continuous control space (by utilizing steer functions instead of a discrete set of randomly generated control propagations) to exploit as much knowledge of the nonholonomic constraints as possible. Brunnen *et al.* [7] propose a two-phase motion planner where A* finds a geometrically feasible path, which then biases the tree growth of RRT*. This planner is applied only to a high-dimensional holonomic robot, where the RRT* vertices (sampled from a Gaussian distribution centered around the A* path) are connected using motion interpolation. In contrast, Theta*-RRT focuses on more complex nonholonomic systems and uses steer functions. Cowlagi and Tsiotras [9] propose a planner that constructs a discrete control set using expensive model-predictive control techniques. In contrast, Theta* adopts a continuous control space. Rickert *et al.* [10] propose the EET planner for holonomic systems that sacrifices probabilistic completeness by using workspace information to continuously adjust the sampling behavior of the planner. In contrast, Theta*-RRT is probabilistically complete.

III. Combining Any-Angle Search with RRT

Let $\mathcal{X} \subset \mathbb{R}^d$ be the state space, $\mathcal{U} \subset \mathbb{R}^m$ the control space, and $\mathcal{X}_{obs} \subset \mathcal{X}$ and $\mathcal{X}_{free} = \mathcal{X} \setminus \mathcal{X}_{obs}$ the obstacle and free spaces, respectively. A (control) system Σ on state space $\mathcal{X}$ is a differential system such that

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t))\, u + g(\mathbf{x}(t)) \qquad \mathbf{x}(0) = \mathbf{x}_{init}, \qquad (1)$$

where $\mathbf{x}_{init} \in \mathcal{X}$ and, for all t, $\mathbf{x}(t) \in \mathcal{X}$ and $\mathbf{u}(t) \in \mathcal{U}$. g describes the drift, and f describes the system dynamics. Theta*-RRT is a feasible motion planner for small-time controllable nonholonomic systems: It finds controls $\mathbf{u}(t) \in \mathcal{U}$ for $t \in [0, T]$ such that the unique trajectory $\mathbf{x}(t)$ that satisfies Equation (1) connects a given start state $\mathbf{x}_{init} \in \mathcal{X}_{free}$ to a given goal state $\mathbf{x}(T) = \mathbf{x}_{goal} \in \mathcal{X}_{goal} \subset \mathcal{X}_{free}$ in the free space $\mathcal{X}_{free}$.

A. Geodesic Distance for Nonholonomic Wheeled Robots

Let us consider small-time controllable nonholonomic systems.

Definition 1: System Σ is locally controllable from $\mathcal{X}$ if the set of states reachable from $\mathcal{X}$ by an admissible trajectory contains a neighborhood of $\mathcal{X}$. It is small-time controllable from $\mathcal{X}$ if, for any time T, the set of states reachable from $\mathcal{X}$ before time T contains a neighborhood of $\mathcal{X}$.

For small-time controllable nonholonomic wheeled robots, we define the geodesic distance $D_{\mathbf{P}}(\mathbf{x}_1, \mathbf{x}_2)$ of two states $\mathbf{x}_1$ and $\mathbf{x}_2$ to a path $\mathbf{P}$ through $\mathbb{R}^2 \times \mathbb{S}^1$. Consider a path $\mathbf{P}$ and let $\mathbf{x}'_1$ and $\mathbf{x}'_2$ be the orthogonal projections of $\mathbf{x}_1$ and $\mathbf{x}_2$ onto $\mathbf{P}$ and their Euclidean distances be $d_1 = \|\mathbf{x}_1 - \mathbf{x}'_1\|$ and $d_2 = \|\mathbf{x}_2 - \mathbf{x}'_2\|$ (respectively). Then, the geodesic distance $D_{\mathbf{P}}(\mathbf{x}_1, \mathbf{x}_2)$ is the sum of the lengths of the geodesics from each of the two states to path $\mathbf{P}$, that is,

$$\begin{aligned} D_{\mathbf{P}}(\mathbf{x}_1, \mathbf{x}_2) \quad = \quad & w_e\,(d_1 + d_2) + w_\theta\left(1 - |\mathbf{q}_{\mathbf{x}_1} \cdot \mathbf{q}_{\mathbf{x}'_1}|\right) \\ & + w_\theta\left(1 - |\mathbf{q}_{\mathbf{x}_2} \cdot \mathbf{q}_{\mathbf{x}'_2}|\right) \end{aligned}$$

(for parameters w_e and w_θ), where $\mathbf{q}_{\mathbf{x}_1}$ and $\mathbf{q}_{\mathbf{x}_2}$ are the quaternions of states $\mathbf{x}_1$ and $\mathbf{x}_2$, and $\mathbf{q}_{\mathbf{x}'_1}$ and $\mathbf{q}_{\mathbf{x}'_2}$ the quaternions of the segments of path $\mathbf{P}$ to which $\mathbf{x}'_1$ and $\mathbf{x}'_2$ belong. The geodesic distance of two states is the smaller, the closer they are to path $\mathbf{P}$ in Euclidean distance, heading orientations and steering orientations.

B. Our Technique: Theta*-RRT

Theta*-RRT (detailed in Algorithm 1) first generates a geometrically feasible any-angle path $\mathbf{P}$ using only geometric information about the workspace. Then, it computes the trajectory by growing a tree τ of smooth

Algorithm 1 *Theta*-RRT*

```
function Theta*-RRT(x_init , x_goal)
P ⇐ AnyAngleSearch(x_init , x_goal)
if P = ∅ then
  return failure
end if
τ.AddNode(x_init)
g(x_init) ⇐ 0
k ⇐ 1
while k ≤ MAX_ITERATIONS do
  x_rand ⇐ AnyAngleSampling(X, P)
  x_near ⇐ NearestNeighborSearch(τ, x_rand, P)
  u_new, σ_new ⇐ Steer(x_near, x_rand)
  if σ_new ∈ X_obs then
    continue
  end if
  τ.AddNode(x_rand)
  τ.AddEdge(x_near, x_rand, u_new)
  g(x_rand) ⇐ g(x_near) + C(x_near, x_rand)
  if x_rand ∈ X_goal then
    return ExtractTrajectory(x_rand)
  end if
  k ⇐ k + 1
end while
return failure
```

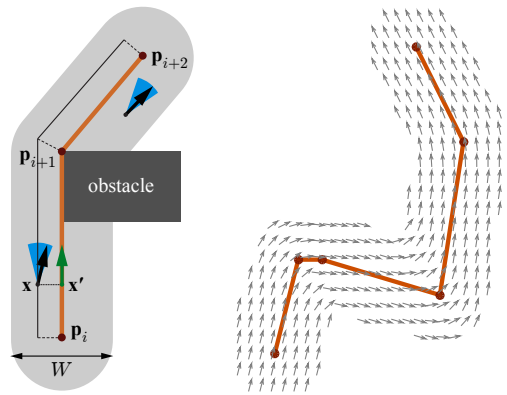

Fig. 3. Path-biased sampling strategy. **Left:** Example strip (in **grey**) around an any-angle path (in **orange**), in which samples are randomly generated. **Black** arrows are samples, and **green** arrows are their projections onto the any-angle path. **Blue** sectors are the angular ranges from which the sample orientations are drawn. The mean sector orientations are computed as weighted averages of the orientations of the any-angle path segments. The individual weight contributions are evaluated in geodesic path coordinates along the offset black line. **Right:** Resulting mean orientations around an example any-angle path.

local geodesics around path $\mathbf{P}$ (path-biasing heuristic) satisfying the system's nonholonomic constraints. It repeatedly samples a state $\mathbf{x}_{rand}$ mainly from a subspace $\mathcal{X}_{local} \subset \mathcal{X}_{free}$ centered around path $\mathbf{P}$. It then makes $\mathbf{x}_{rand}$ a new tree vertex and connects it to $\mathbf{x}_{near}$, which is selected among several ones as the vertex that connects with minimum cost to $\mathbf{x}_{rand}$. The cost depends on the length and smoothness of the trajectory from the candidate tree vertex to state $\mathbf{x}_{rand}$ and the geodesic distance of both vertices to the any-angle path. The subroutines of Algorithm 1 are described below:

`AnyAngleSearch`($\mathbf{x}_{init}$, $\mathbf{x}_{goal}$) uses Theta* to search an eight-neighbor grid from start grid vertex s_{init} to goal grid vertex s_{goal}, where S is the set of all grid vertices. $s_{init} \in S$ is the grid vertex that corresponds to the start vertex $\mathbf{x}_{init}$, and $s_{goal} \in S$ is the grid vertex that corresponds to the goal vertex $\mathbf{x}_{goal}$. We assume obstacles cells to be inflated so as to reflect the robot shape. Theta* uses the consistent straight-line distances as heuristics. It returns an any-angle path $\mathbf{P} = \{\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_N\}$ (a discrete Cartesian path) if one exists and the empty path otherwise.

`AnyAngleSampling`($\mathcal{X}, \mathbf{P}$) samples mainly from a connected subspace $\mathcal{X}_{local} \subset \mathcal{X}$ according to a distribution that conveys geometric information of path $\mathbf{P}$ and returns the sampled state $\mathbf{x}_{rand}$. Concretely, the Cartesian components of the samples are generated uniformly from a strip with width W (for parameter W, called position bias) centered around path $\mathbf{P}$. To convert the any-angle path into a smooth trajectory, the heading orientation x_θ and steering orientation x_δ of the samples are generated uniformly from angular intervals centered around a mean orientation $\bar{\alpha}$, which is a linear combination of the orientations of the segments of path $\mathbf{P}$, that is,

$$\bar{\alpha} = \sum_{i=1}^{N} w_i \, \alpha^i_{\mathbf{p}}, \tag{2}$$

where $\alpha^i_{\mathbf{p}}$ is the orientation of segment $\overline{\mathbf{p}_i\mathbf{p}_{i+1}}$. The weights w_i are calculated from trapezoidal membership functions that are associated with each segment. The functions are centered around the centers of their segments with tails that overlap into the neighboring segments such that their values at the path vertices $\mathbf{p}_i$ are exactly 0.5 and their slopes are no less than a minimal slope δ_S (for parameter δ_S). The influence of each membership function on a given sample $\mathbf{x}$ is computed along geodesic path coordinates, obtained by offsetting path $\mathbf{P}$ with the perpendicular distance of $\mathbf{x}$ to $\mathbf{P}$ (see Fig. 3, left). The orientations x_θ and x_δ of the samples are then generated uniformly from the interval $(\bar{\alpha} - \Delta\theta, \bar{\alpha} + \Delta\theta)$ (for parameter $\Delta\theta$, called orientation bias). The components of the samples that are not related to the workspace (such as velocities and accelerations) are generated uniformly. Moreover with a frequency $f_{uniform}$, this function generates uniformly a sample from the entire $\mathcal{X}_{free}$ (for parameter $f_{uniform}$).

`NearestNeighborSearch`(τ, $\mathbf{x}_{rand}$, $\mathbf{P}$) returns the tree vertex $\mathbf{x}_{near}$ that connects with minimum cost $C(\mathbf{x}_{near}, \mathbf{x}_{rand})$ to state $\mathbf{x}_{rand}$. Instead of determining tree vertex $\mathbf{x}_{near}$ directly, Theta*-RRT determines a set of tree vertices $\mathcal{X}_{near}$ within distance δ_R from $\mathbf{x}_{rand}$ (for parameter δ_R). If this set is empty, it returns the tree vertex nearest to $\mathbf{x}_{rand}$. Otherwise, it returns the tree vertex from set $\mathcal{X}_{near}$ that connects with minimum cost $C(\mathbf{x}_{near}, \mathbf{x}_{rand})$ to state $\mathbf{x}_{rand}$, that is,

$$\mathbf{x}_{near} = \arg\min_{\mathbf{x} \in \mathcal{X}_{near}} C(\mathbf{x}, \mathbf{x}_{rand}) \tag{3}$$

with

$$C(\mathbf{x}, \mathbf{x}_{rand}) = g(\mathbf{x}) + C_\sigma + D_{\mathbf{P}}(\mathbf{x}, \mathbf{x}_{rand}), \tag{4}$$

where $g(\mathbf{x})$ is the sum of the costs from the tree root $\mathbf{x}_{init}$ to the tree vertex $\mathbf{x}$ and $D_{\mathbf{P}}(\mathbf{x}, \mathbf{x}_{rand})$ the geodesic distance of states $\mathbf{x}$ and $\mathbf{x}_{rand}$ from path $\mathbf{P}$. The cost C_σ measures the length and smoothness of the trajectory σ from tree vertex $\mathbf{x}$ to state $\mathbf{x}_{rand}$ returned by the steer function. It is defined as

$$C_\sigma = \sum_{i=0}^{N_e-1} w_d ||\sigma_{i+1} - \sigma_i|| + w_q \left(1 - |\mathbf{q}_{i+1} \cdot \mathbf{q}_i|\right)^2$$

(for parameters w_d and w_q), where $N_e + 1$ is the number of intermediate states σ_i on trajectory σ and $\mathbf{q}_i$ are the associated quaternions. The cost C_σ can be computed on-line or very efficiently with a regression approach [11].

`Steer`($\mathbf{x}_{near}$, $\mathbf{x}_{rand}$) returns controls $\mathbf{u}_{new}$ and a trajectory σ_{new} from state $\mathbf{x}_{near}$ to state $\mathbf{x}_{rand}$ with terminal time T. The analytical steer function connects any pair of states and respects the *topological property* [12], that is, for any $\epsilon > 0$ there exists some $\eta > 0$ such that, for any two states $\mathbf{x}_{near} \in \mathcal{X}$ and $\mathbf{x}_{rand} \in \mathcal{X}$ with

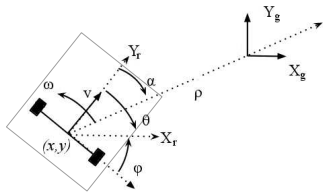

Fig. 4. Differential drive system in polar coordinates: ρ is the Euclidean distance between the Cartesian coordinates of the robot pose (x, y, θ) and of the goal state, ϕ the angle between the x-axis of the robot reference frame $\{X_r\}$ and the x-axis of the goal state frame $\{X_g\}$, α the angle between the y-axis of the robot reference frame and the vector connecting the robot with the goal position, v the translational and ω the angular robot velocity.

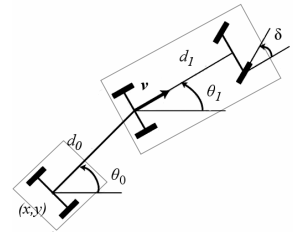

Fig. 5. Truck-and-trailer system: $(x, y) \in \mathbb{R}^2$ are the coordinates of the trailer axle's midpoint, θ_0 and θ_1 the orientations of the trailer and truck, v the translational velocity of the truck, δ its steering angle, d_1 the distance between the front axle and the rear axle of the truck, and d_0 the distance between the trailer axle and the hitch joint on the rear truck axle.

$||\mathbf{x}_{near} - \mathbf{x}_{rand}|| < \eta$, it holds that $||\mathbf{x}_{near} - \sigma_{new}(t)|| < \epsilon$ for all $t \in [0, T]$. If σ_{new} is collision-free, it is added to τ as the tree branch (or edge) that connects $\mathbf{x}_{near}$ to $\mathbf{x}_{rand}$.

IV. Experimental Setup

We now investigate how well Theta*-RRT performs against the baseline planners RRT, A*-RRT, RRT* and A*-RRT*. All planners extend their trees using steer functions. RRT and RRT* sample in the entire state space. RRT uses C_σ as distance metric, and RRT* uses C_σ as cost function. A*-RRT and A*-RRT* sample along A* paths. A*-RRT generates samples and selects the tree vertex that connects to the sampled state with minimum cost in the same way as Theta*-RRT. A*-RRT* generates the samples from a Gaussian distribution centered around the A* path as in [7] and uses C_σ as cost function.

All experiments are carried out with a C++ implementation on a single core of an ordinary PC with a 2.67 GHz Intel i7 processor and 10 GB RAM. The weights of the cost function and the parameters of the distance metric are $w_d = w_e = w_q = w_\theta = 0.5$ and $\delta_S = \delta_R = 4m$. $f_{uniform}$ is set to 1 over 5000 samples.

A. Nonholonomic Systems

We consider two small-time controllable nonholonomic systems, namely a 3-dimensional differential drive system and an 8-dimensional truck-and-trailer system.

Differential drive system: We use a unicycle system with state (x, y, θ), where $(x, y) \in \mathbb{R}^2$ is the Cartesian position and $\theta \in [-\pi, \pi)$ is the heading orientation. After a Cartesian-to-polar coordinate transformation, see Fig.4, the equations of motions are

$$\begin{aligned} \dot{\rho} &= -\cos\alpha \; v \\ \dot{\alpha} &= \frac{\sin\alpha}{\rho} v - \omega \\ \dot{\phi} &= -\omega, \end{aligned} \tag{5}$$

where v and ω are the translational and the angular velocities, respectively. For this system, we use the efficient and smooth steer function POSQ [13]. Width and length of the robot are set to $0.4m$ and $0.6m$, respectively.

Truck-and-trailer system: In order to obtain high-quality trajectories, we use the extended state $(x, y, \theta_0, \theta_1, v, \dot{v}, \delta, \dot{\delta})$, where $(x, y) \in \mathbb{R}^2$ are the coordinates of the trailer axle's midpoint, θ_0 and θ_1 the orientations of the trailer and truck, respectively, v the translational velocity of the truck, $\dot{v}$ its acceleration, δ the steering angle of the truck and $\dot{\delta}$ its derivative, see Fig. 5. The equations of motions are

$$\begin{aligned} \dot{x} &= v\cos(\theta_1 - \theta_0)\cos\theta_0 \\ \dot{y} &= v\cos(\theta_1 - \theta_0)\sin\theta_0 \\ \dot{\theta}_0 &= \frac{v}{d_0}\sin(\theta_1 - \theta_0) \\ \dot{\theta}_1 &= \frac{v}{d_1}\ \tan(\delta). \end{aligned} \tag{6}$$

For this system, we use the η^4 splines [14] as steer function since they are known to generate high-quality trajectories for truck-and-trailer systems. We set $d_0 = d_1 = 1$, width and length of the trailer to $0.4m$ and $0.6m$ and the truck width to $0.4m$.

B. Environments

To stress-test the planners and study how they behave in environments of varying complexity, we design three simulated test environments shown in Figs. 1 and 6. The *maze* environment in Fig. 1 contains many different homotopy classes, has local minima (such as U-shaped obstacles) and narrow passages. Its size is 50m×50m. The *random* environment contains randomly generated square obstacles, its size is 50m×30m. The *narrow corridor* environment in Fig. 6 stresses the ability of the planners to generate smooth trajectories in narrow corridors, its size is 25m×25m. The grid cell size for the any-angle search is 1 m in all environments.

C. Performance Metrics

For each planner and environment, we perform 100 runs for the differential drive system and 50 runs for the truck-and-trailer system. We are solely interested in the first trajectories found. We compute the means and standard deviations of the following four performance metrics for all planning problems that are solved within the planning time limit of 1,000 seconds: tree size N_v (measured in the number of stored tree vertices), planning time T_s (measured in milliseconds or seconds) and resulting trajectory length l_p (measured

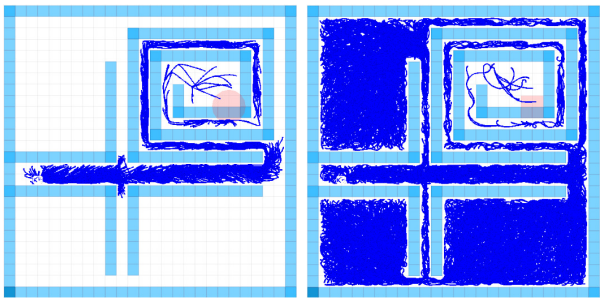

Fig. 6. *Narrow corridor* environment with the goal position (in **red**) and the trees (in **blue**). **Left:** Tree of Theta*-RRT. **Right:** Tree of RRT. Theta*-RRT generates a smaller tree than RRT, which makes Theta*-RRT faster.

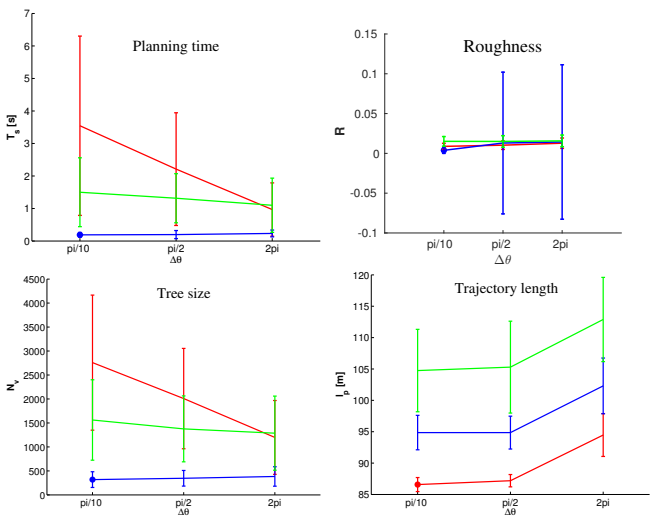

Fig. 7. Performance trends for different strengths of the position bias W ($W = 1m$ in **red**, $W = 4m$ in **blue** and $W = 10m$ in **green**) and orientation bias $\Delta\theta$ in the *maze* environment for the metrics planning time, roughness, tree size and trajectory length (smaller values are better for all performance metrics).

in meters). Smoothness, although an intuitive concept, is less straightforward to assess. In our previous work [11, 13], we used performance metrics based on the velocity profile of the robot (such as the average speed arc lengths, velocity profile peaks or normalized jerk). Here, we use a metric that is better suited for measuring *geometric* trajectory smoothness and thus human-perceived smoothness (namely how sharp the turns are): roughness R, defined as the square of the change in curvature κ of the robot, integrated along the trajectory and normalized by the trajectory length L,

$$R = \int_{t_0}^{t_l} \left| \frac{1}{L} \frac{d\kappa}{dt} \right|^2 dt.$$

A smaller roughness indicates smoother trajectories. We also compute the percentage of trajectories found (problems solved) within the planning time limit.

D. Theta*-RRT Parameters

Prior to the main experiment, we analyze the impact of the parameters W and $\Delta\theta$ on the performance of Theta*-RRT. Position bias W is related to the geometry of the wheeled robot and should be set to a value no less than the maximum value of its length and width. We use the *maze* environment and the ranges $W = \{1m, 4m, 10m\}$ and $\Delta\theta = \left\{\frac{\pi}{10}, \frac{\pi}{2}, 2\pi\right\}$. For each pair of parameter values, we compute the mean and standard deviation of the four performance metrics over multiple runs. Fig. 7 shows the results for the differential drive system. The results for the truck-and-trailer system are qualitatively similar. We observe three trends: *(i)* With a larger orientation and position bias (that is, smaller $\Delta\theta$ and W), the trajectories tend to be shorter and smoother, which is expected since the trajectories then follow the any-angle paths more closely. *(ii)* With a smaller orientation bias (that is, larger $\Delta\theta$), the tree sizes and planning times tend to be smaller. The optimum is at the medium value $W = 4m$ where the value of $\Delta\theta$ has almost no influence (but the optimum is at the smallest value $\Delta\theta = \frac{\pi}{10}$). Given these trends, we select the medium position bias $W = 4m$ and the strong orientation bias $\Delta\theta = \frac{\pi}{10}$.

V. Experimental Results

The experimental results for Theta*-RRT and the four baseline planners are given in Tables I-II. Smaller values are better for all performance metrics. The best values are highlighted in boldface. Theta*-RRT outperforms the four baselines with respect to all performance metrics, with only two exceptions. It is a close second with respect to trajectory smoothness to RRT* for the differential drive system in the *random* environment and to A*-RRT for the truck-and-trailer system in the *narrow corridor* environment. We make the following observations:

(i) The path-biasing heuristic of Theta*-RRT avoids the time-consuming exploration of the entire state space and thus results in small tree sizes and planning times. This advantage comes at the cost of having to find an any-angle path first but Tab. III shows that the runtime of the discrete search is negligible compared to the overall planning time. Theta*-RRT thus has an advantage over RRT and RRT* that explore large parts of the state space, especially in environments with local minima and narrow passages. For this reason, RRT* (and even A*-RRT*) fail to find any trajectory within 1,000 seconds for the high-dimensional truck-and-trailer system in all runs in two of the three environments.

(ii) The path-biasing heuristic of Theta*-RRT results in trajectories that fall into good homotopy classes and are thus short. Theta*-RRT thus has an advantage over A*-RRT and A*-RRT*, whose path-biasing heuristics suffer from the A* paths typically being in worse homotopy classes than the Theta* paths, which results in longer trajectories and thus also larger planning times and tree sizes.

(iii) The sampling strategy of Theta*-RRT results in smooth trajectories. Theta*-RRT thus has an advantage over A*-RRT*, whose sampling strategy is not quite as sophisticated.

Random environment									
Planner	Tree size $\mathbf{N_v}$		Planning time $\mathbf{T_s}$ $[s]$		Trajectory length $\mathbf{l_p}$ $[m]$		Roughness $\mathbf{R}$		Problems solved
Theta*-RRT	54	$\pm$ 59	**0.011**	$\pm$ 0.007	**43.11**	$\pm$ 1.485	0.001	$\pm$ 0.003	100%
A*-RRT	49	$\pm$ 46	0.020	$\pm$ 0.01	44.16	$\pm$ 1.76	0.003	$\pm$ 0.004	100%
RRT	137	$\pm$ 150	0.09	$\pm$ 0.05	65.25	$\pm$ 13.54	0.009	$\pm$ 0.008	100%
RRT*	168	$\pm$ 154	9.57	$\pm$ 13.73	43.84	$\pm$ 1.45	**0.00074**	$\pm$ 0.00140	100%
A*-RRT* [7]	**32**	$\pm$ 34	0.40	$\pm$ 0.93	52.88	$\pm$ 19.0	0.0057	$\pm$ 0.0098	100%
Maze environment									
Planner	Tree size $\mathbf{N_v}$		Planning time $\mathbf{T_s}$ $[s]$		Trajectory length $\mathbf{l_p}$ $[m]$		Roughness $\mathbf{R}$		Problems solved
Theta*-RRT	**319**	$\pm$ 164	**0.19**	$\pm$ 0.07	**94.86**	$\pm$ 2.74	**0.0038**	$\pm$ 0.0038	100%
A*-RRT	1470	$\pm$ 777	3.73	$\pm$ 4.8	98.45	$\pm$ 1.12	0.015	$\pm$ 0.007	100%
RRT	2615	$\pm$ 960	4.85	$\pm$ 4.62	139.16	$\pm$ 21.63	0.018	$\pm$ 0.01	100%
RRT*	658	$\pm$ 37	16.13	$\pm$ 0.34	129.61	$\pm$ 5.65	0.024	$\pm$ 0.01	100%
A*-RRT* [7]	356	$\pm$ 193	47.66	$\pm$ 48.47	96.57	$\pm$ 5.37	0.013	$\pm$ 0.009	100%
Narrow corridor environment									
Planner	Tree size $\mathbf{N_v}$		Planning time $\mathbf{T_s}$ $[s]$		Trajectory length $\mathbf{l_p}$ $[m]$		Roughness $\mathbf{R}$		Problems solved
Theta*-RRT	**1206**	$\pm$ 258	**8.16**	$\pm$ 3.63	**77.78**	$\pm$ 1.44	**0.0027**	$\pm$ 0.004	100%
A*-RRT	1799	$\pm$ 715	18.84	$\pm$ 15.2	77.8	$\pm$ 1.33	0.023	$\pm$ 0.01	100%
RRT	8488	$\pm$ 1639	180.45	$\pm$ 58.55	78.36	$\pm$ 1.82	0.0069	$\pm$ 0.005	100%
RRT*	45310	$\pm$ 7012	2667.5	$\pm$ 481.7	79.47	$\pm$ 0.9	0.03	$\pm$ 0.008	100%
A*-RRT* [7]	3236	$\pm$ 572	309.4	$\pm$ 119.4	78.37	$\pm$ 0.65	0.0125	$\pm$ 0.006	100%

TABLE I

Experimental results: Trajectory quality and planning efficiency for the differential drive system.

Random environment									
Planner	Tree size $\mathbf{N_v}$		Planning time $\mathbf{T_s}$ $[s]$		Trajectory length $\mathbf{l_p}$ $[m]$		Roughness $\mathbf{R}$		Problems solved
Theta*-RRT	**52.2**	$\pm$ 48.3	**0.0547**	$\pm$ 0.0790	**44.331**	$\pm$ 2.8418	**0.0057**	$\pm$ 0.0060	100%
A*-RRT	75.7	$\pm$ 52.4	0.1019	$\pm$ 0.0984	51.74	$\pm$ 7.89	3.4993	$\pm$ 8.8502	58%
RRT	836	$\pm$ 378	1.32	$\pm$ 0.84	66.96	$\pm$ 14.7	2.17	$\pm$ 2.00	100%
RRT*	3957	$\pm$ 2756	816.16	$\pm$ 656.58	52.39	$\pm$ 13.12	0.54	$\pm$ 1.01	76%
A*-RRT* [7]	3582	$\pm$ 3138	949.6	$\pm$ 823.7	49.30	$\pm$ 12.79	0.1013	$\pm$ 0.2647	100%
Maze environment									
Planner	Tree size $\mathbf{N_v}$		Planning time $\mathbf{T_s}$ $[s]$		Trajectory length $\mathbf{l_p}$ $[m]$		Roughness $\mathbf{R}$		Problems solved
Theta*-RRT	**522**	$\pm$ 167	**2.57**	$\pm$ 1.50	**98.59**	$\pm$ 4.95	**1.0073**	$\pm$ 0.7226	100%
A*-RRT	661	$\pm$ 181	4.56	$\pm$ 2.0858	101.79	$\pm$ 8.26	1.1317	$\pm$ 1.0372	100%
RRT	4858	$\pm$ 1276	38.88	$\pm$ 15.83	126.34	$\pm$ 16.52	2.0788	$\pm$ 1.2985	100%
RRT*	0	$\pm$ 0	0	$\pm$ 0	0	$\pm$ 0	0	$\pm$ 0	0% – **failed**
A*-RRT* [7]	0	$\pm$ 0	0	$\pm$ 0	0	$\pm$ 0	0	$\pm$ 0	0% – **failed**
Narrow corridor environment									
Planner	Tree size $\mathbf{N_v}$		Planning time $\mathbf{T_s}$ $[s]$		Trajectory length $\mathbf{l_p}$ $[m]$		Roughness $\mathbf{R}$		Problems solved
Theta*-RRT	**1513**	$\pm$ 492	**20.87**	$\pm$ 13.77	**77.10**	$\pm$ 6.75	2.15	$\pm$ 1.12	100%
A*-RRT	2139	$\pm$ 573	33.46	$\pm$ 16.74	79.66	$\pm$ 5.94	**1.9352**	$\pm$ 0.8722	100%
RRT	1794	$\pm$ 5473	733.98	$\pm$ 438.28	83.77	$\pm$ 7.04	2.31	$\pm$ 1.47	100%
RRT*	0	$\pm$ 0	0	$\pm$ 0	0	$\pm$ 0	0	$\pm$ 0	0% – **failed**
A*-RRT* [7]	0	$\pm$ 0	0	$\pm$ 0	0	$\pm$ 0	0	$\pm$ 0	0% – **failed**

TABLE II

Experimental results: Trajectory quality and planning efficiency for the truck-and-trailer system.

Additionally, we tested Theta*-RRT in a real-world setting by deploying it on a passenger guidance robot for complex and busy airport environments (Fig. 8).

VI. Probabilistic Completeness of Theta*-RRT

The results clearly demonstrate the benefit of Theta*-RRT. However, its path-biasing heuristic – as any heuristic – can mislead and even degrade the performance of RRT, for example when the any-angle path is infeasible to follow under kinodynamic constraints, although a geometric solution (in the inflated grid world) exists. In such cases, the probabilistic completeness, a key property of RRT, is lost. In this section, we prove that Theta*-RRT retains the probabilistic completeness for all small-time controllable nonholonomic systems which use an analytical steer function. Probabilistic completeness is well established for systems with geometric constraints [15] and kinodynamic systems under some strong assumptions (that is, forward simulations [2], uniform sampling and optimal steering [16, 17] and holonomic systems with state-space based interpolation [18]). Our proof follows the one introduced in [2] but we consider a special class of nonholonomic systems, namely systems that are *small-time controllable*, see Definition 1.

Environments	$\mathbf{T_{Theta^*}}$ $[ms]$	$\mathbf{T_{A^*}}$ $[ms]$
Random	5.34	8.07
Maze	12.06	19.91
Narrow corridor	45.14	37.74

TABLE III

Experimental results: Planning times of Theta* and A*

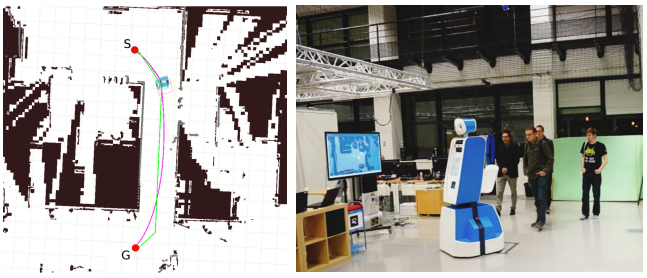

Fig. 8. Theta*-RRT on a real differential drive robot. **Right:** The robot guides a group of people. **Left:** The dots S and G (in **red**) represent the start and goal positions (respectively). The any-angle path (in **green**) is generated first, followed by the smooth trajectory (in **purple**).

Theorem 1: Consider a small-time controllable non-holonomic system. Define a non-zero and non-uniform continuous sampling distribution f_s over $\mathcal{X}_{free}$ generated by the path-biasing technique. Let Theta*-RRT use an analytical steer function that connects any pair of states in $\mathcal{X}$. Then, Theta*-RRT is probabilistically complete since the probability of connecting the start state $\mathbf{x}_{init} \in \mathcal{X}_{free}$ to the goal state $\mathbf{x}_{goal} \in \mathcal{X}_{free}$, if possible, approaches one asymptotically.

Proof: Let $\mathcal{B}(\mathbf{x}_i, \rho)$ denote the ball of radius $\rho > 0$ centered on $\mathbf{x}_i \in \mathcal{X}_{free}$. Consider all the tree vertices $\cup_{i=0,..,k}\ \mathbf{x}_i \in \tau$ at iteration k. Since the volume $\Omega = \cup_{i=0,..,k}\ \mathcal{B}(\mathbf{x}_i, \rho \geq \delta_R > 0)$ is non-zero for the Lebesgue metric the event of sampling a state $\mathbf{x}_{rand} \in \Omega$ will happen with probability one as the number of iterations goes to infinity. Given that the system is small-time controllable, the connection (performed by the *steer* function) of $\mathbf{x}_{rand}$ to $\mathbf{x}_{near}$ (chosen among multiple vertices in τ), will be successful and therefore (if collision free) $\mathbf{x}_{rand}$ will be added to τ. The set $\tilde{\mathcal{X}}_k = \{\mathbf{x} \in \mathcal{X}_{free} \backslash \forall \mathbf{x} \in \tau\}$ represents the uncovered part of the space $\mathcal{X}_{free}$ by τ. By induction following the above property, as k approaches infinity, $\mu(\tilde{\mathcal{X}}_k)$ (the volume of $\tilde{\mathcal{X}}_k$) approaches zero, therefore the state $\mathbf{x}_{goal}$ will be added to τ with probability one. ■

Theorem 1 extends to RRT with any path-biasing heuristic as long as it uses analytical steer functions for systems that are small-time controllable since the proof does not exploit any geometric properties of $\mathcal{X}_{local}$.

VII. Conclusions

In this paper, we introduced Theta*-RRT, a hierarchical technique that combines (discrete) any-angle search with (continuous) RRT motion planning for small-time controllable nonholonomic wheeled robots. We evaluated the approach using two different non-holonomic systems in three different environments and compared it to four different baseline planners, namely RRT, A*-RRT, RRT* and A*-RRT*. The results show that Theta*-RRT finds shorter trajectories significantly faster than the baselines without loss of smoothness, while A*-RRT* and RRT* (and thus also Informed RRT* [8]) fail to generate a first trajectory sufficiently fast in environments with complex nonholonomic constraints. We also proved that Theta*-RRT retains the probabilistic completeness of RRT for all small-time controllable systems that use an analytical steer function.

Acknowledgments

The authors thank Aurelio Piazzi for providing access to the η^4 splines closed-form expressions. This work has been supported by the EC under contract number FP7-ICT-600877 (SPENCER). Sven Koenig's participation was supported by NSF under grant numbers 1409987 and 1319966 and a MURI under grant number N00014-09-1-1031. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the sponsoring organizations, agencies or the U.S. government.

References

[1] K. Daniel, A. Nash, S. Koenig, and A. Felner, "Theta*: Any-angle path planning on grids," *Artificial Intelligence Research, Journal of*, vol. 39, no. 1, 2010.

[2] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," *Int. Journal of Robotics Research*, vol. 20, 2001.

[3] S. Karaman and E. Frazzoli, "Incremental sampling-based algorithms for optimal motion planning," in *Robotics: Science and Systems (RSS)*, Zaragoza, Spain, 2010.

[4] E. Plaku, L. E. Kavraki, and M. Y. Vardi, "Discrete search leading continuous exploration for kinodynamic motion planning," in *Robotics: Science and Systems (RSS)*, Philadelphia, USA, 2007.

[5] E. Plaku, E. Kavraki, and M. Y. Vardi, "Motion planning with dynamics by a synergistic combination of layers of planning," *Robotics, IEEE Transactions on*, vol. 26, no. 3, 2010.

[6] K. Bekris and L. Kavraki, "Informed and probabilistically complete search for motion planning under differential constraints," in *First International Symposium on Search Techniques in Artificial Intelligence and Robotics (STAIR), Chicago, IL*, 2008.

[7] M. Brunner, B. Bruggemann, and D. Schulz, "Hierarchical rough terrain motion planning using an optimal sampling-based method," in *Int. Conf. on Robotics and Automation (ICRA)*, Karlsruhe, Germany, 2013.

[8] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Informed RRT*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic," in *Int. Conf. on Intelligent Robots and Systems (IROS)*, Chicago, USA, 2014.

[9] R. V. Cowlagi and P. Tsiotras, "Hierarchical motion planning with dynamical feasibility guarantees for mobile robotic vehicles," *Robotics, IEEE Transactions on*, vol. 28, no. 2, 2012.

[10] M. Rickert, A. Sieverling, and O. Brock, "Balancing exploration and exploitation in sampling-based motion planning," *Robotics, IEEE Transactions on*, vol. 30, no. 6, 2014.

[11] L. Palmieri and K. O. Arras, "Distance metric learning for RRT-based motion planning with constant-time inference," in *Int. Conf. on Robotics and Automation (ICRA)*, Seattle, USA, 2015.

[12] J.-P. Laumond, S. Sekhavat, and F. Lamiraux, *Guidelines in nonholonomic motion planning for mobile robots.* Springer, 1998.

[13] L. Palmieri and K. O. Arras, "A novel RRT extend function for efficient and smooth mobile robot motion planning," in *Int. Conf. on Intelligent Robots and Systems (IROS)*, Chicago, USA, 2014.

[14] F. Ghilardelli, G. Lini, and A. Piazzi, "Path generation using η^4-splines for a truck and trailer vehicle," *Automation Science and Engineering, IEEE Transactions on*, vol. 11, no. 1, 2014.

[15] A. M. Ladd and L. E. Kavraki, "Measure theoretic analysis of probabilistic path planning," *Robotics and Automation, IEEE Transactions on*, vol. 20, no. 2, 2004.

[16] D. Hsu, R. Kindel, J.-C. Latombe, and S. Rock, "Randomized kinodynamic motion planning with moving obstacles," *Int. Journal of Robotics Research*, vol. 21, 2002.

[17] E. Frazzoli, M. A. Dahleh, and E. Feron, "Real-time motion planning for agile autonomous vehicles," in *American Control Conference*, vol. 1, 2001.

[18] S. Caron, Q.-C. Pham, and Y. Nakamura, "Completeness of randomized kinodynamic planners with state-based steering," in *Int. Conf. on Robotics and Automation (ICRA)*, Hong Kong, China, 2014.